Q1. Describe three applications for exception processing.

ANS: **Try and except statements** are used to catch and handle exceptions in Python. Statements that can raise exceptions are kept inside the try clause and the statements that handle the exception are written inside except clause.1

It can consist of 3 steps: **a try block that encloses the code section which might throw an exception**, one or more catch blocks that handle the exception and. a finally block which gets executed after the try block was successfully executed or a thrown exception was handled. There are three types of exception—**the checked exception, the error and the runtime exception**.

|  |
| --- |
| # Python program to demonstrate finally    # No exception Exception raised in try block  try:      k = 5//0  # raises divide by zero exception.      print(k)    # handles zerodivision exception  except ZeroDivisionError:      print("Can't divide by zero")    finally:      # this block is always executed      # regardless of exception generation.      print('This is always executed') |

**Output-:**

Can't divide by zero

This is always executed

Q2. What happens if you don't do something extra to treat an exception?

Ans:- When an exception occurred, if you don't handle it, **the program terminates abruptly and the code past the line that caused the exception will not get executed**.

Q3. What are your options for recovering from an exception in your script?

Ans:-

1. A try-catch-finally statement is a code or program that handles exceptions.
2. The try clause runs the code that generates exceptions.
3. The catch clause catches exceptions that are thrown.
4. A finally clause always gets executed.
5. The throw statement generates exceptions.

Q4. Describe two methods for triggering exceptions in your script.

Ans:- There are two methods to handle Python exceptions:

**Try** – This method catches the exceptions raised by the program.

Raise – Triggers an exception manually using custom exceptions.

As a Python developer you can choose to throw an exception if a condition occurs. To throw (or raise) an exception, **use the raise keyword**.

Q5. Identify two methods for specifying actions to be executed at termination time, regardless of whether or not an exception exists.

Ans:- A finally block must be associated with a try block, you cannot use finally without a try block. ... In normal case when there is no exception in try block then the finally block is executed after try block. However if an exception occurs then **the catch block is executed before finally block**.

CATCH:-

try

}

catch (ArrayIndexOutOfBoundsException e) {

System.err.printin("Caught first " + e.getMessage()); } catch (IOException e) {

System.err.printin("Caught second " + e.getMessage());

}

FINALLY:-

try:

raise KeyboardInterrupt

finally:

print 'welcome, world!'